|  |
| --- |
| Sutro Games  Solar Siege |
| Design Document |
|  |
| **Josh Becker**  **Ryan Bone** |
| **Jack Craft** |
| **Ed Grudner** |
| **Navid Matin** |
| **Hector Medina-Fetterman** |
| **Chase Plante** |

Table of Content:

[Introduction](#h.esjmxdvgz3lt)

[Rules](#h.uuwb64gzxsvs)

[Win Condition](#h.h3v8cxhulgqf)

[Resources](#h.z4v5pxw9jzn5)

[Towers](#h.wxz8l6bpzsg4)

[Game Flow](#h.27hqpvjctsmh)

[Story](#h.qkxaurpjezg8)

[Towers](#h.9fsts8vrym8x)

[Construction](#h.hlsrdqivy5lm)

[Control](#h.buaonl8asuf)

[Tower Types](#h.aggw3fz784ac)

[Sniper Tower](#h.yobt8ob5d6ir)

[Rocket Tower](#h.8vybpocwgr65)

[Machine-Gun Tower](#h.vjkgxnu2bj5d)

[Poison](#h.3hxc4pkaljr)

[Betrayal](#h.z5np0gtye0rv)

[Slow](#h.n7rkdz1s34tm)

[Fear](#h.9sre8tufoh3c)

[Abilities](#h.bcy5nxyrxck3)

[Knockback](#h.ql8faz3vn3ki)

[Repair Tower](#h.i1zxg3bknvt0)

[Recycle Tower](#h.f1jvfy9zq27v)

[Portal](#h.lwzt04skims3)

[Enemies](#h.j4qnwgrwr1hx)

[Bat-Cat](#h.xp3ot3rdtv8h)

[Angry Mushrooms!](#h.5qozgroqaxj8)

[3. Toad Trolls! (Coming Soon)](#h.w057zgr7e4k1)

[Enemy AI](#h.sj0thpl6r3ur)

[Planets](#h.sfr9fyxhk3m3)

[Planet Features](#h.uh8hwrrfcfsq)

[Enemy Spawners](#h.yaalhq8ugw7j)

[Shields and Generators](#h.q7csrltb0ovn)

[Enemy Towers](#h.hfm5ebg5taz5)

[Resource Pools](#h.iwzjwi3xzibw)

[Tutorial Planet:](#h.yhzdebwvobv6)

[Bat Cat Planet:](#h.63ncxdvmbijy)

[Toad Planet:](#h.e3d72ojsuuyy)

[Fungi Planet:](#h.dslrjtm5k3ty)

[Controls:](#h.1qv2ijw5y4g4)

[Code Design:](#h.cbmafpqebk7n)

[Managers:](#h.zbwr1wdi277u)

[Script Behavior:](#h.tedmbyvaq658)

# Introduction

Solar Siege is the result of taking the common concept of the tower-defense and reversing it. The player character’s attacks are still done through placing towers; however rather than defending against hordes of enemies, the player instead becomes the aggressor, repeatedly encroaching on enemy territory until the opponent is defeated.

# Rules

## Win Condition

Victory is achieved by destroying the enemy fortress.

## Resources

The player can obtain resources by defeating enemies, or selling turrets.

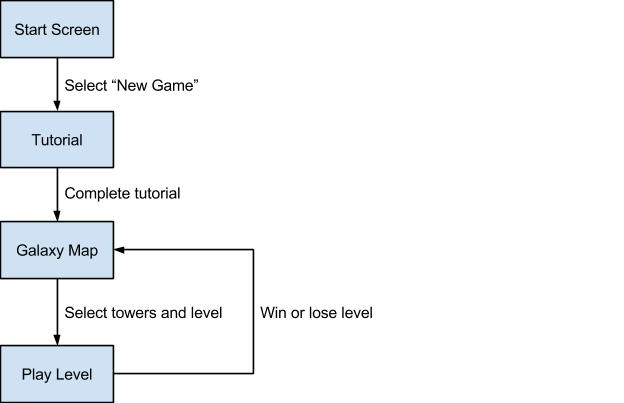
## Towers

Players have the option to choose up to 4 towers before entering each planet. Players obtain more towers by conquering enemy planets and stealing the inhabitant’s tower technology.

**Abilities**

Players are supplied with four abilities which include a knockback skill, the ability to heal structures, the ability to sell structures, and a portal ability.

# Game Flow



# Story

The main character is Lt. Jack Hammer. Or at least he was a lieutenant while he was in the earthly space army, before he realizes that he can make more money by joining the Corpyco.. Corpyco has one product, alien pets! After the emergence of new alien mind control technology, Corpyco. was able to conquer alien planets and sell them as pets. You get assigned a new solar system on the first day of the job. As a perk you also get a nice bunny infested planet to keep your resources on in that solar system. Now you only have one task, to conquer all of these aliens and ship them to earth so Corpyco can sell them at a nice price and keep you paid!

# Towers

## Construction

The player’s main way to interact with enemies in the game is by building towers, each consisting of a base and a turret. Players have a variety of tower types to choose from which can be combined in any way. Some tower types include the sniper, rocket and machine gun tower. A player can use any available types as either the base or the turret in order to build a unique tower. A tower’s statistics and abilities are determined by its included types. The turret determines the base damage, auto fire range and rate of fire. The base contributes a multiplier to each of these statistics.

Each tower type has a cost associated with it. A tower’s total cost is determined by its two types. When building towers the player must take into account how much each part costs.

Some tower types, like the rocket type, contribute a secondary effect. A rocket turret’s projectiles will directly damage whatever they hit and anything within a given area around the point of impact. When a type like rocket is included in a tower this secondary effect is always applied to the tower’s projectiles. A rocket base with a machine gun tower will shoot fairly quickly due to the machine gun’s fast rate of fire and also provide the area of effect damage from the rocket base.

The ability to combine any two types into a single tower gives the player a large number of options from a fairly small pool of tower types.

Before entering a level the player gets to choose four tower types to bring with them. The player gets to assign these types to A, B, X, and Y as they see fit. While on the planet the player can press any one of these buttons and confirm it with right trigger to create a tower in front of their character consisting only of that type. The player can also press two of these buttons sequentially to create a base/turret pair where the first button is the base and the second is the turret. The player can then confirm that pair. If the player is unhappy with the current pair, they can press another button and the current base/turret pair will update by pushing the current base out, replacing the previous base with the previous turret and adding the newly selected tower type as the current turret. The player can then confirm this new pair with the right trigger. If at any time the player does not want to build any tower, but they have pressed any combination of A, B, X, or Y, they may press the left bumper to cancel the build.

We use a single script, called *TowerBuilder*, in order to keep track of the current button presses and to actually create the towers in the world. Whenever a player can build a tower and presses one of these buttons, we take note of the button’s associated tower type. We use this type to look up and instantiate the corresponding tower base and turret. We are creating the full tower GameObject, and altering it so that it does not interact with the world. When the player confirms a tower, we alter it again in order to place it in the world and fully activate it.

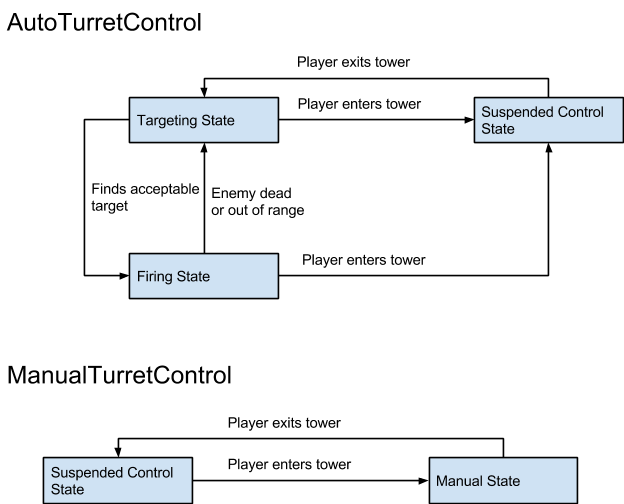
## Control

Towers automatically fire at enemies within a certain range determined by the tower’s statistics. Players can also directly control towers by getting into them. A player may get into a tower by going close to it and clicking the right joystick on the controller. This will transition the camera to a first person view from the tower. While in the tower, the player may aim it with the right joystick and fire with the right trigger.

When the player gets into a tower, the tower is given a boost in all statistics for a short time before the ability stops and must cool down. This short lived boost encourages players to build a variety of towers and move from one to another whenever the boost runs out.

Tower control is implemented with a state system. There are two scripts on each tower that keep track of its state. One script, *AutoTurretControl*, manages the tower when it is automatically finding targets and firing at them and the other, *ManualTurretControl*, manages the tower when the player enters it.

*AutoTurretControl* works by simply starting the tower in a targeting state, suspending control when the player enters the tower, and starting up in a targeting state when the player exits. Targeting is done with a function called “OverlapSphere” which takes a position and a radius and returns all objects within that sphere. We look through those objects to determine if there are any acceptable targets. Each tower has a list of tags that it will shoot at. A target is deemed acceptable if its tag matches one of these tags and is not dead. We choose the closest acceptable target and then switch to the firing state. While in the firing state, a tower will track the target and shoot as often as it can, as long as the tower has a clear shot at the target. The tower will return to the targeting state when the target dies or goes out of range.



## Tower Types

|  |  |  |
| --- | --- | --- |
| Sniper Tower High damage  Low speed  Long range  No secondary effect | Rocket Tower High damage  Low speed  Medium range  Area damage and knockback effect | Machine-Gun Tower Low damage  High speed  Short range  No secondary effect |
| Poison Low damage  Medium speed  Medium range  Damage over time effect   Betrayal Low damage  Medium speed  Medium range  Causes enemies to fight for you for a short time | Slow Low damage  Medium speed  Medium range  Slows enemy movement speed | Fear Low damage  Medium speed  Medium range  Reverses enemy movement |

# **Abilities**

The player has four abilities that can be selected using the d-pad on a controller or number keys 5-8 on a keyboard. All of the abilities share a cool down of 2 seconds, meaning, the player can only use an ability once every 2 seconds.

## Knockback

Knockback applies a force to enemies that are in front of the player, pushing them away so that the player can make an escape and avoid being overwhelmed. This ability is selected by pressing down on the d-pad of a controller or by pressing ‘5’ on a keyboard.



## Repair Tower

Using this while facing a tower will repair the tower, increasing its health by 10. This ability is selected by pressing left on the d-pad of a controller or by pressing ‘6’ on a keyboard.

## 

## Recycle Tower

Recycling a tower will destroy the tower and return ⅔ of the credits that it cost to the player. This ability is selected by pressing up on the d-pad of a controller or by pressing ‘8’ on a keyboard.

## 

## Portal

The player can place two portals around the world and travel between them at will. Creating a third portal with destroy the oldest of the previous pair so only two portals are active at one time. This ability is selected by pressing right on the d-pad of a controller or by pressing ‘7’ on a keyboard.

# **Enemies**

Each enemy and boss is specific for a planet. As the player continues the game and goes through more levels he/she will face new enemies and bosses. By defeating a planet the player can use the conquered enemy as a pet.

## Bat-Cat

Location: Bat Cat Planet

## 

## Angry Mushrooms!

Location: Fungi Planet

Special Abilities: Mind Control / Fear



## 3. Toad Trolls! (Coming Soon)

Location: Toad Planet!

Special Abilities: Strong close range attack, don’t get too close!

# Enemy AI

Our artificial intelligence for enemies consists of three behavior archetypes which are managed on each planet by a manager class. All enemies will notify the enemy manager of any player units they encounter.

**Scouts**

Scouts are primarily tasked with seeking out the player and his towers. This is accomplished by placing specific nodes on the planet which scouts arbitrarily travel between, marking targets they see along the way. Scouts do not attack, and are the weakest of the archetypes, but are also the fastest.

**Sentries**

Sentries defend the enemy’s interests on the map, such as the spawners, generators, and towers. When spawned sentries query the ai manager for a patrol which is a list of adjacent nodes, the sentry then will navigate to its patrol and then being patrolling, both looped and unlooped patrols are supported. When sentries encounter an enemy they will attack it, however they refuse to stray beyond a specific distance from their patrols. This archetype has the highest health, but is slower and has weaker attacks.

**Brawlers**

Brawlers seek out the player’s units and destroy them. When brawlers spawn they query the ai manager for a node where an enemy was recently spotted, then they travel to that location. If a brawler encounters one of the player’s units then they will pursue it until it kills the target or is killed itself. Brawlers have moderate health, higher attack, and moderate speed.

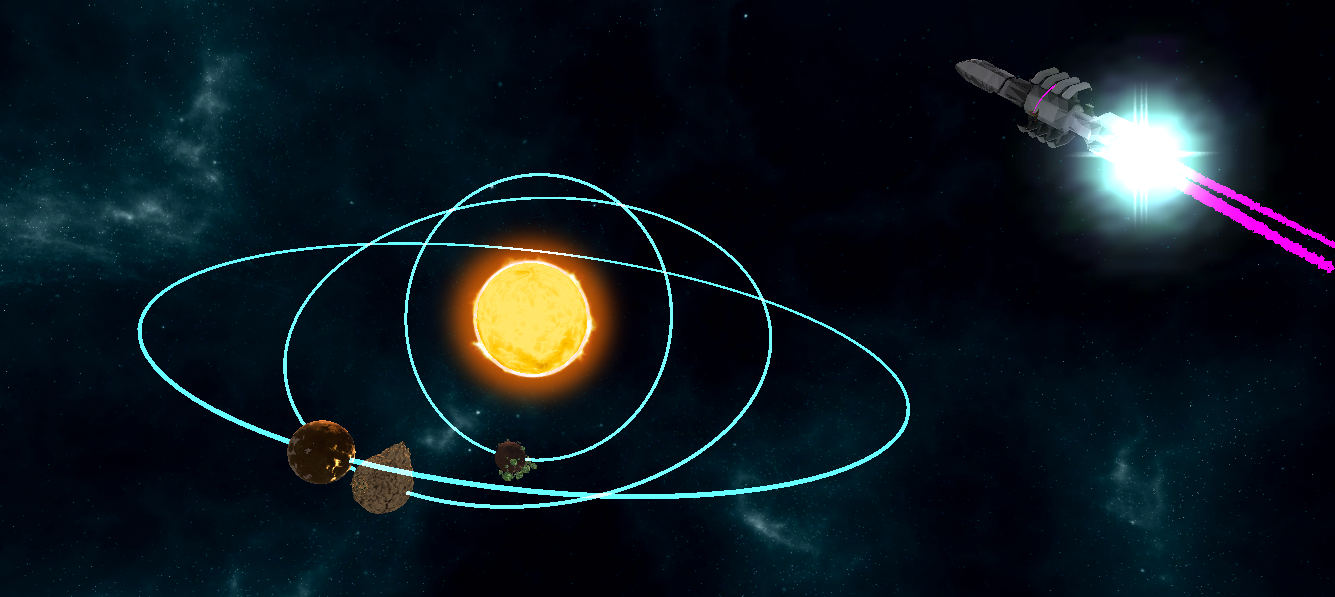
**Navigation**

Our enemies navigate using a custom made NavMesh which consists of nodes strewn about the planet. Each node contains a list of adjacent nodes, creating a directed graph which enemies use to navigate. When an enemy wishes to travel they are supplied a path, which is a list of adjacent nodes, by the AI manager. Enemies then travel node by node until they reach their destination.



World

Our game world takes place in a fantasy galaxy far out in the fringes of space.



# Planets

# Planet Features

### Enemy Spawners

Planets are populated by various types of spawners spread across the planets.

### Shields and Generators

Some planets have large protective shields surrounding enemy towers and spawners. These shields have inordinate amounts of health and players should not try to take them down directly. Players rather should search the world for the generator powering the shield. When the generator is destroyed, the shield goes away.

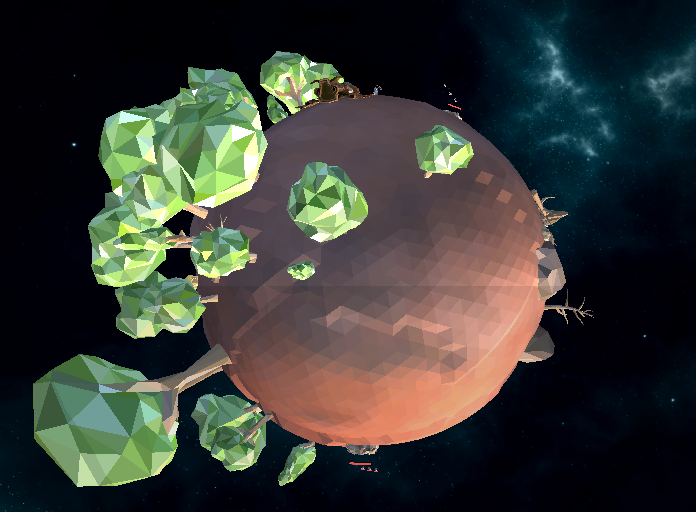
### Enemy Towers

The player is up against not only roving enemies, but also stationary towers that defend the spawners and generators.

### Resource Pools

Some planets have resources that the player can mine for additional wealth.

## Tutorial Planet:



In this planet, players learn the basics of game play and have a simple mission to destroy rabbits which cannot hurt the player.

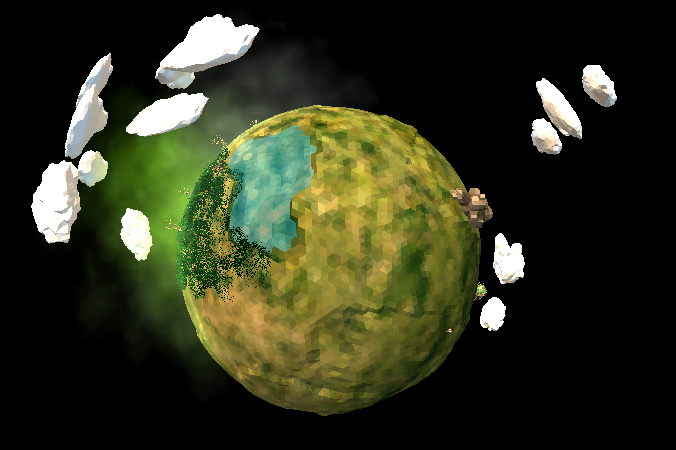
This is the first level of the game and serves as a tutorial.

## Bat Cat Planet:



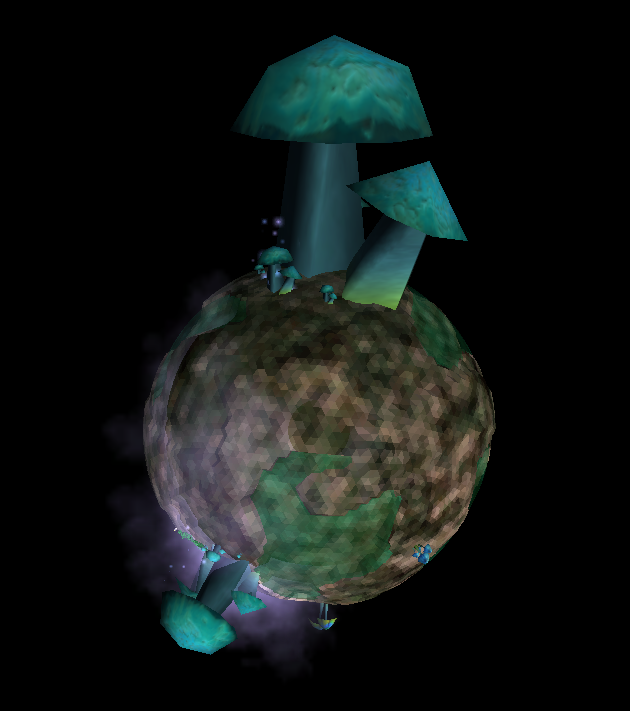
This is the first actual level that player can lose. In this level the player fights enemies which are hybrid bat-cats complete with a shield and generator which must be defeated in order to conquer the planet. The bat-cat planet is an arid and hot environment, with its deserts and mountains. However it seems like a perfect place for mining, and the many ruins may yield safe havens for players, and maybe even a surprise or two.

## Toad Planet:



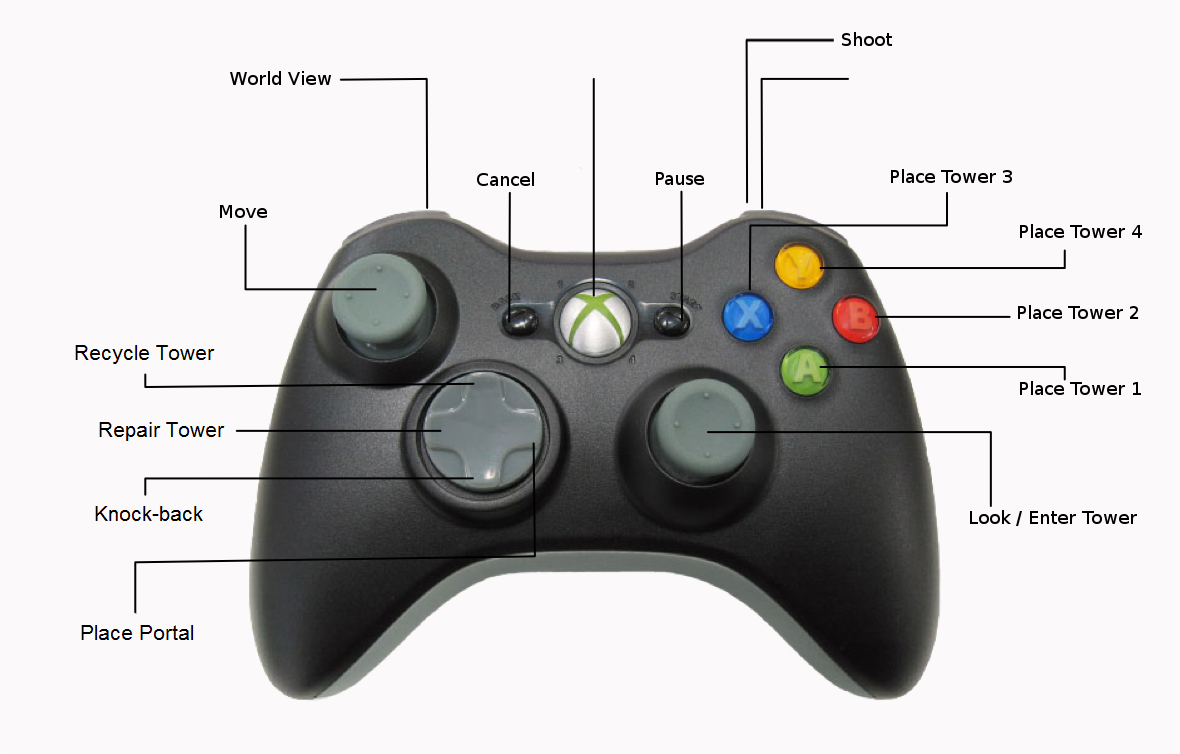
The toads’ planet is characterized by fog-covered lakes, and its troll infested mountains, both of which are teeming with hostile inhabitants. However between these two biomes are open grasslands where the player can begin their invasion.

## Fungi Planet:



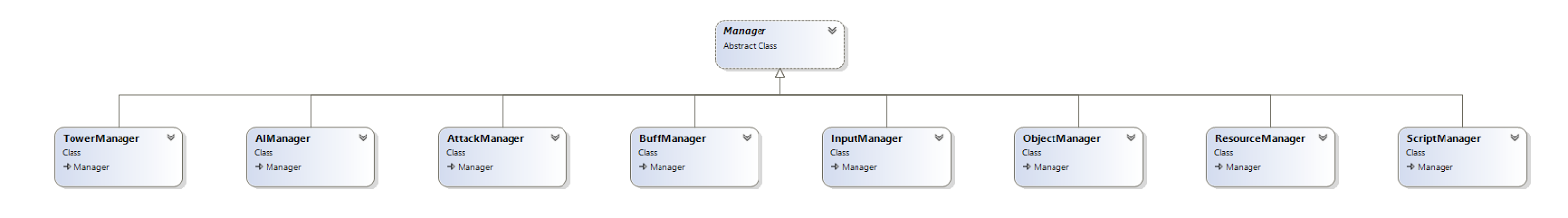
The fungi planet is wet and covered in smog. Years of culmination in this dank humidity has yielded massive mushrooms, some of which have achieved sentience and may fight back. However mushrooms are not the only hostile force to watch out for, this planet has a plethora of flora which will leave you at the mercy of their spores.

# Controls:



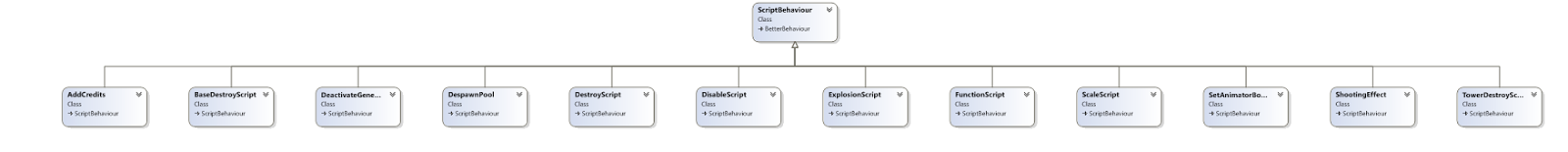
# Code Design:

## Managers:



1. Tower Manager
2. AI Manager
3. Attack Manager
4. Buff Manager
5. Input Manager
6. Object Manager
7. Resource Manager
8. Script Manager

## Script Behavior:



1. AddCredits
2. BaseDestroy
3. DeactiveGenerator
4. DespawnPool
5. Destroy
6. Disable
7. Explosion
8. Function
9. Scale
10. SetAnimatorBoolean
11. ShootingEffect
12. TowerDestroy

User Interface

Tower UI



The tower bar shows which towers the player has selected at the galaxy map and shows the player which buttons map to which tower archetype.

Health UI



The health bar lets the player keep track of their current health.

Credit UI



The credit UI lets the player keep track of their remaining credits.

Player Abilities UI



This lets the player see which abilities are available to them, and which ability is currently active. The bottom ability is a knock-back, the left ability is tower repair, the top ability lets the player recycle placed towers, and the right ability lets the player place a portal.

Dialog

Solar Siege uses Unity 4.6 for its GUI and Dialog. The dialog system allows for custom trigger zones to trigger dialog cut scenes. The flexibility of the system allows for many unique applications, such as interact/use prompts and dialog between characters.

Dialog is broken into multiple components. The Dialog component stores data to be placed into the GUI. The Hierarchy of data is as follows:

* + Dialog
    - Dialog Tree
      * Dialog Frames
        + Dialog Buttons

By having multiple trees we allow for the possibility of probabilistic dialog that changes randomly through gameplay.

